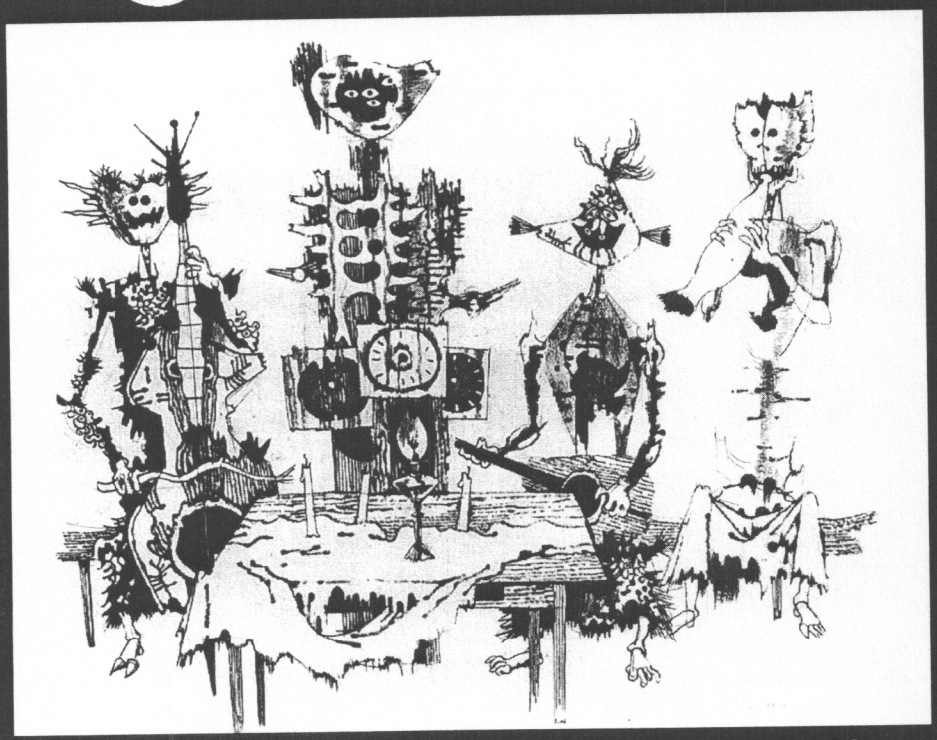
Speech and Image Understanding

IAPR International Association for Pattern Recognition



Alexander von Humboldt Foundation

Edited by: PAVEŠIĆ, NIEMANN, KOVAČIČ, MIHELIČ

IEEE Slovenia Section

N. Pavešić, H. Niemann, S. Kovačič, F. Mihelič (Eds.)

Speech and Image Understanding

Proceedings of 3rd Slovenian-German and 2nd SDRV Workshop
April 24-26, 1996, Ljubljana, Slovenia

IEEE Slovenia Section

Editors

Nikola Pavešić
Faculty of Electrical Engineering, University of Ljubljana, Ljubljana, Slovenia

Heinrich Niemann
Lehrstuhl für Mustererkennung, Universität Erlangen-Nürnberg, Erlangen, Germany

Stanislav Kovačič
Faculty of Electrical Engineering, University of Ljubljana, Ljubljana, Slovenia

France Mihelič
Faculty of Electrical Engineering, University of Ljubljana, Ljubljana, Slovenia

CIP - Kataložni zapis o publikaciji
Narodna in univerzitetna knjižnica, Ljubljana

681.3.019(063)
519.243(063)

SLOVENSKO društvo za razpoznavanje vzorcev. Workshop (2 ; 1996 ; Ljubljana)
Speech and image understanding: proceedings of 3rd Slovenian-German and 2nd SDRV Workshop, April 24 - 26, 1996, Ljubljana, Slovenia / eds. N. Pavešić ... [et.al.]. - [Ljubljana] : IEEE Slovenia section, 1996

ISBN 961-6062-10-7

1. Gl. stv. nasl. 2. Pavešić, Nikola. - I. SDRV. Workshop (2 ;
1996 ; Ljubljana) glej Slovensko društvo za razpoznavanje vzorcev.

Workshop (2 ; 1996 ; Ljubljana)
62075136

Cover picture: France Mihelič sr., Musicians and Clocks, (charcoal, 70 × 100cm), 1974.

Printing and binding: SOMARU d.o.o. Ljubljana, Slovenia
Printed in Slovenia

Po mnenju Mininstrstva za znanost in tehnologijo Republike Slovenije št. 415-01-105-96 z dne 18. 9. 1996 gre za proizvod, od katerega se plačuje davek od prometa proizvodov po tarifni številki 3.

Speech and Image Understanding, Proceedings of 3rd Slovenian-German and 2nd SDRV Workshop, April 24-26, 1996, Ljubljana, Slovenia, pp. 199–207.

Searching for Faces in Image Data Bases Using Machine Learning

Janez Demšar and Franc Solina

Computer Vision Laboratory

Faculty of Computer and Information Science

University of Ljubljana

Tržaška 25, SI-1001 Ljubljana, Slovenia

E-mail: janez.demsar@snet.fri.uni-lj.si, franc.solina@fri.uni-lj.si

Abstract

While there are simple and fast techniques for querying conventional databases and collections of texts, content-based retrieving of images still remains a problem to be solved. To eliminate the need for a trained user and refining queries, needed by some recent systems, we propose the use of machine learning to induct decision trees as a substitute for conventional queries. We did experiments on querying for pictures of faces using machine learning algorithm ID3 and the results are promising.

1 Introduction

Large data storage media and ever growing worldwide networking have led to generation of large collections of texts, images and other types of data. While there are simple, fast and useful methods for locating a specific text, content-based retrieving of images is still a problem to be solved. Existing systems for content based image retrieving (CBIR) generally use attributes that are manually or automatically extracted from images and then stored and managed in conventional database systems [3]. Since they require a trained user, they cannot be used by nontechnical staff. Besides that, pre-calculated attributes are often too domain specific or too general. Chabot System [5] for example, integrates a relational database containing keyword and other conventional data with color analysis technique to allow searching by keywords and dominant colors. It allows queries as "*mostlyOrange* and *someBlue*" which should, presumably, describe images of sunsets over seas and lakes. The problem with this approach is in finding the right combination of attributes; query must be often refined. Also, those queries do not seem to describe the content of the image accurately enough.

To eliminate the need for a trained operator and refining queries we propose to use a query by example images technique. To search for an image, user has to

provide some positive and negative examples and the program learns to distinguish between them.

The next subsection takes a closer look at machine learning by examples technique. The rest of the article is divided in three sections. Section 2 describes experiments using simple attributes in image retrieving by examples, in particular domain specific attributes for retrieving images of faces. Section 3 experiments using domain specific attributes which the learner system can find by itself, enabling therefore retrieval of images of any contents. Conclusions are in the last section.

1.1 Machine learning

Learning from examples is a form of learning where "the teacher" provides a series of examples and "the learner" makes generalizations about them. Gathered knowledge can be represented in several different ways (*if-then* rules, semantic networks, decision trees). We decided to use a modified algorithm ID3 [6] from the family of top-down induction of decision tree (TDIDT) algorithms. Given examples described by a (finite) set of numeric and/or nonnumeric attributes and a class it belongs to, the algorithm builds a decision tree. Each node contains a binarized attribute (for example, "the amount of the black color is between 25 and 45.7 percent") and each leaf contains the probability that an example, classified to that leaf, belongs to a certain class.

Basic ID3 works like this:
if *all the examples belong to the same class C*
the result is a leaf labeled C
else
choose the most informative attribute A
divide examples according to the value of A
recursively build subtrees for subpopulations

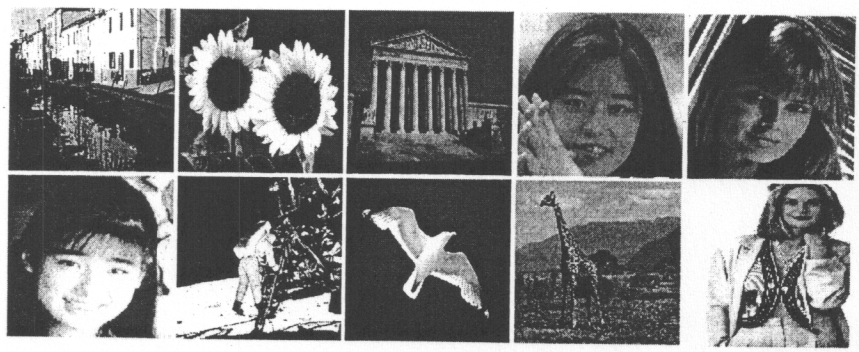


Figure 1: Some examples of images from our database.

Beside simple construction methods, the reason for popularity of decision trees also lies in their efficiency and transparency. In some domains, especially in medicine, it is desirable to know what reasoning does the computer use to make a decision. The same holds for CBIR: by knowing how a learning algorithm uses attributes, we can see which attributes are useless and which are more important so they should be refined. An important advantage of decision trees over non-transparent knowledge presentation forms is also the user's ability to manually edit it in case he wants to incorporate some properties that learning algorithm had not discovered ("learning by example" can therefore be combined by "learning by being told"). For further discussion on decision trees see references [1, 4, 7].

1.2 Experimental setup

We worked with a collection containing 197 scanned photographs of different sizes in eight-bit RGB format. 69 images represented a human face of bright complexion (i.e. the face occupied a reasonable part of the image) while other pictures had different contents (Figure 1). The program had to learn to distinguish pictures of faces from the rest of the pictures.

We approached the problem in two phases:

In the first phase, we defined a set of simple domain dependent attributes, extracted them from images and fed the collected data to a general purpose machine learning program Magnus Assistant [2]. Results demonstrated that the chosen attributes describe the domain accurate enough to be useful in CBIR.

In the second phase, we excluded domain dependent attributes and wrote a general tool for CBIR. Instead of using the Assistant we implemented a problem specific machine learner which is given only simple domain independent attributes and some general data about images. When domain independent attributes showed to be useless, the learner was expected to invent domain specific attributes by itself.

Both methods were tested on the same database so that the results can be compared.

2 Manually specified attributes

In this chapter we describe the first phase of our work; we list attributes that were used for describing images and examine the properties of decision trees build by Assistant by using those attributes.

2.1 The set of attributes

In order to write a useful image retrieving system all the attributes involved in decision trees have to be simple, since there is no time to calculate sophisticated attributes on all images from a large collection. Attributes that we decided to use for our experiment of querying for pictures of faces are

- **proportions of basic colors (white, black, green, blue, yellow and red) on the whole image;** each point of the image is approximated to one of this colors using a Manhattan distance.
- **proportions of basic colors in the middle of the image; The middle of the image** is defined to be the rectangular area covering points that are at least $\frac{1}{5}$ of width and height away from borders of the image (Figure 2). For pictures of objects, like faces, fruits or vehicles, proportions of colors in that part of the picture turned out to be more important than proportions of colors on the whole picture. However, that does not hold for pictures of nature, satellite shots and other scenic pictures.

Figure 2: Definition of the middle of the image.

- **similarity of colors in the middle and in the whole image** is defined as $\sum_C p_C \cdot p'_C / \sqrt{\sum_C p_C^2 \cdot \sum_C p'^2_C}$, where p_C is proportion of the color C and p'_C proportion of the color C in the middle of the picture.
- **proportion of edge points;** boundary extraction is simplified and therefore inaccurate.
- **proportion of color of the skin on the whole image and in the middle;** the white skin color $S = (R = 185, G = 125, B = 100)$ was determined manually. A color $C = (r, g, b)$ is similar to skin color if it satisfies the following conditions:

$$d_\phi(S, C) = 1 - \frac{S \circ C}{|S||C|} \leq 0.005 \tag{1}$$

$$d_{lum} = ||S| - |C|| \leq 100 \tag{2}$$

  where $S \circ C$ is a scalar product and $|C|$ is $\sqrt{C \circ C}$.
- **distribution of skin color** is measured by means of average and dispersion over columns and rows and by coefficients, obtained with interpolation of distribution function by Fourier series.

The last two attributes are obviously domain dependent. All attributes are simple enough to be computed in a single pass over the image. Besides that, in most of the popular graphics formats there is no need to keep the entire image in the working memory. If necessary, memory consumption can be limited to only two rows of picture at once.

2.2 Building and testing trees

After extracting attributes' values, general knowledge elicitation tool Magnus Assistant was used to build and test decision trees. 70% of examples were used as learning examples and the rest as test examples. Learning examples were chosen at random and the decision tree was built many times by using different set of learning images.

Decision trees built by ID3 correctly classify all learning examples. Because of presence of noise it usually makes more sense to prune the trees; pruning makes the trees smaller and also more accurate since it cuts off the noise. The prunning method we used, was to stop building the tree when a majority class in a node exceeded 90% of population. When some node has, for example, 20 images and 19 of them represent a human face it seems plausible to ignore the twentieth.

The quality of decision trees was measured by classification accuracy and by tree size which has direct impact on the speed of searching.

2.3 Results

Analysis of attributes. As expected, the proportion of **skin color** in the middle of the picture was by far the most important attribute. In all experiments Assistant found out that examples with less than 9 or 10% of skin color do not represent a human face. More surprisingly, experiments showed a proportion of the **blue color** to be the second most important attribute; almost all the trees classified the image as a non-face if it contained too much (around 1 %) of blue color in the middle. The most probable reason is that the skin is approximated with white, yellow or red, besides that there could be some hair in the middle of the image, so the only two basic colors that are normally not appearing in the middle area are green and blue. The distribution of skin color was rarely used. It always appeared close to leaves and was often pruned off. The other approach to fast shape observation, the edge points proportion, appeared only in one of 20 experimental runs. We can therefore conclude that attributes, dealing with proportions of colors are much more useful than (simple) shape describing attributes.

Size and quality of generated trees. Average tree contained 6 internal nodes. In average, generated trees (prunned at $p = 90\%$) correctly classified 88.1% of ("previously unseen") test examples.

Some generated trees. The smallest generated tree (Figure 3 contains only two nodes: the image represents a face if it contains more than 10% of skin color and less than 1% of blue in the middle. Classification accuracy, as measured on test examples, was 86.2%.

Many other trees started the same way. An interesting tree, which is shown in Figure 4 continues by checking the proportion of skin color once again. If there is more than 23% of skin, it is a face, if not, there must be at least 38% of black color. The reason for this unusual condition is, probably, that the color of bright hair is the same as the color of skin. If there is less than 23% of skin color, the person was most probably dark-haired and there is a lot ($\geq$ 38%) of black color.

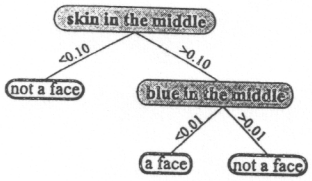


Figure 3: The smallest generated tree.

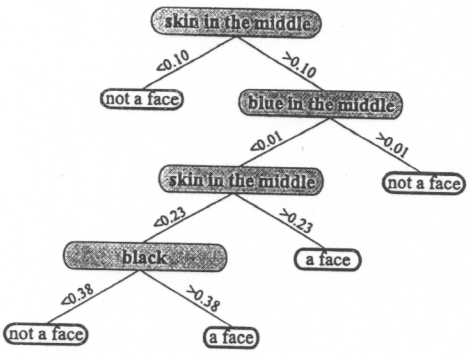


Figure 4: A typical tree.

3 The domain independent learner

In the second phase of our experiments we have replaced the general knowledge elicitation tool by our own learner that does not only learn from predefined attributes but is also able to search for new, domain specific attributes, which enables him to serve as a domain independent learner.

3.1 New types of attributes

Color attributes. To reduce the human involvement in the learning process but still to keep the system domain independent, we eliminated the need for specifying most descriptive color(s). The "amount of skin color" attribute was eliminated but the learner was given ability to find it by itself. To accomplish this, it uses local optimization technique with informativity as a criterion function.

Success of optimization depends upon the size of the population. For a large population, the criterion function is smooth and the local optimization method can give good results. But during the building of a decision tree the size of the population can decrease to just a few examples; in this case more random methods, like simulated annealing, would probably give better results. On the other hand, it makes no sense to spend time by discovering attributes to classify the last few examples that are, most probably, just results of noise and will not contribute to the tree's accuracy.

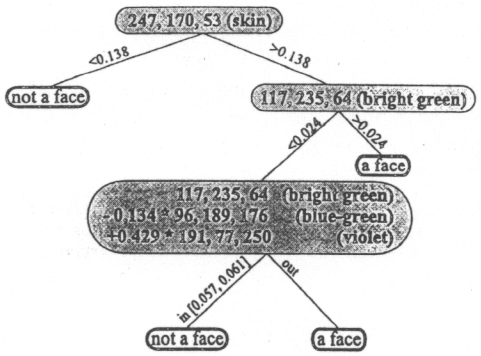


Figure 5: One of the most accurate trees

The solution that we used in our learner program is *the inheritance of attributes.* When searching for useful color attributes, the program does not only invent new attributes but also tries to further optimize some of the most informative attributes that have been found and optimized in previous step of tree induction.

Composed attributes. After finding some good color attributes, the program composes an attribute whose value is a linear combination of values of the best found attribute and the two most correlated attributes. Linear coefficients are determined by local optimization.

3.2 Results

Results after building more than 300 decision trees showed that classifying images by means of automated invention of color attributes is even more accurate than classifying by manually defined domain specific attributes.

Inherited attributes are almost always better than new invented attributes. It seems that optimization's success strongly depends upon population size. **Proportions of colors in the middle of the image** are superior to the colors of the whole image. Querying for scenic images would probably inverse the situation. **Composed attributes** do not improve the classification accuracy although they have higher informativity than uncomposed ones. This is obviously a consequence of overfitting the data. **Effective size of decision tree** is the average number of nodes that an unseen example will traverse before being classified. Effective size of generated trees is 1.41, i.e. in average, 1.41 colors have to be calculated to classify an image. Having in mind that real world collections pictures of faces normally present much less than one third of all pictures, we can, observing decision trees, conclude that real effective size is even much lower. **The average classification accuracy** on test examples was 88.6% (the average was calculated on trees, generated with optimal number of inherited attributes and without composed attributes).

One of the best trees is shown in Figure 5. The program "discovered" the color of the skin; almost 14 percent of the image must be covered by this color in order to recognize picture as a face. The rest is the same as before, blue color is replaced

by green and a composed attribute is added to refine the result. The classification accuracy of this tree on test examples is 94.9%.

The learning program often used composed attributes to express conditions like "*there must be a lot of skin color in the middle but not on the whole image*". This hint can help it to construct precise and extremely small trees (Figure 6).

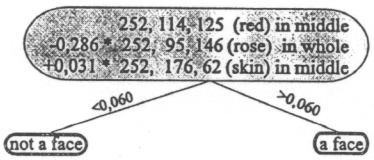


Figure 6: A very small tree using a composed attribute.

4 Conclusions

In our work we show that decision trees are a promising way to deal with the problem of content-based retrieving of images. An important advantage of our approach is its simplicity. The user is not required to have any technical insight and there is no need to refine queries. However, it is possible to do so by showing the learner some of misclassified images, forcing him to try to improve the tree (incremental learning).

Color attributes found by domain independent learner are much more accurate than basic colors' attributes used by some other systems. Also, attribute values' ranges calculated by binarization algorithm are more exact than manually defined artificial ranges. Searching for an image is fast.

Disadvantage of the program is the low speed of learning. Since it uses a relatively slow local optimization method to find optimal attributes, the process of learning can take few minutes; however, a decision tree that is calculated only once can be used many times. Also, it seems that quite a lot of examples are needed for learning - in order to search for pictures of some kind we must already have a small collection of them.

In the future, we will test the program on other domains; we will try its abilities in searching for images of trees, houses, sunsets and similar. We will also incorporate new types of attributes, including attributes describing texture, structure and shape. Attributes that are computationally more expensive will be limited to occur only in lower levels of the tree, when images are already filtered by faster attributes. By introducing new, slower but better attributes, the user will be able to decide between fast non-accurate and slow but accurate searching.

New attributes might also require images to be reloaded to observe properties that cannot be stored in a precomputed attribute database (for example, after discovering that amount of skin color is an important attribute, the learner program might observe the shape of objects of that color). Low-resolution thumbnails or scalespace techniques can be used for that purpose.

References

[1] Bratko I. (1990), Prolog programming for artificial intelligence, second edition. Addison-Wesley, Wokingham, England.

[2] Cestnik B., Kononenko I., Bratko I (1987) ASISTANT 86: a knowledge elicitation tool for sophisticated users, In *Progress in Machine Learning*, pp. 31-45. Sigma Press, Wilmslow, England.

[3] Gudivada V. N., Raghavan, V., V. (1995) Content-Based Image Retrieval Systems, In *Computer 28*, pp. 18-12.

[4] Niblett T. B., Bratko I. (1986) Learning decision rules in noisy domains, In *Expert Systems 86*. Cambridge University Press.

[5] Ogle V. E. (1995) Chabot: Retrieval from a Relational Database of Images, In *Computer 28*, pp. 40-48.

[6] Quinlan J. T. (1986) Induction of decision trees, In *Machine Learning 1*, pp. 81-106. Kluwer Academic Publishers.

[7] Seidelmann. G. (1993) Using Heuristics to Speed up Induction on Continuous-Valued Attributes, *6th European Conference on Machine Learning*, pp. 390-394. Springer Verlag, Berlin, Germany.